

































-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  **CSS**
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  PL/SQL
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  T-SQL
-  VB.NET
-  VB6
-  XML



CSS static code analysis

Unique rules to find Bugs and Code Smells in your CSS code

All rules 25 Bug 16 Code Smell 9

Tags ▾

Search by name... 🔍

Single line comment syntax should not be used
Bug
CSS properties should be valid
Bug
Units should be valid
Bug
"calc" operands should be correctly spaced
Bug
Color definitions should be valid
Bug
Expressions within "calc" should be valid
Bug
Selectors should be known
Bug
Shorthand properties that override related longhand properties should be avoided
Bug
"linear-gradient" directions should be valid
Bug
"at-rules" should be valid
Bug
Media features should be valid
Bug
Pseudo-element selectors should be valid
Bug
Pseudo-class selectors should be valid
Bug

Pseudo-element selectors should be valid

Analyze your code

Bug Major ?

The W3C specifications define the valid pseudo-element selectors. Only the official and browser-specific pseudo-element selectors should be used to get the expected impact in the final rendering.












Noncompliant Code Example

```
a::beforre { /* Noncompliant; there is a typo on the word before */
...
}
```

Compliant Solution

```
a::before {
...
}
```

Available In: sonarcloud | sonarqube

<div>Properties should not be duplicated</div> <div> Bug</div>
<div>"!important" should not be used on "keyframes"</div> <div> Bug</div>
<div>Font declarations should contain at least one generic font family</div> <div> Bug</div>
<div>CSS files should not be empty</div> <div> Code Smell</div>
<div>Selectors should not be duplicated</div> <div> Code Smell</div>
<div>Empty blocks should be removed</div> <div> Code Smell</div>
<div>Strings should not contain new lines</div> <div> Code Smell</div>
<div>Duplicated font names should be removed</div> <div> Code Smell</div>
<div>Multi-line comments should not be empty</div> <div> Code Smell</div>
<div>Duplicate imports should be removed</div> <div> Code Smell</div>
<div>Extra semicolons should be removed</div> <div> Code Smell</div>
<div>Selectors of lower specificity should</div>